MODÜL 14

SQLite VERİ TABANI

Şekil 14.1: *Bölümle ilgili örnek uygulamalara karekod' dan ulaşabilirsiniz*

Veri tabanları verilerin kalıcı olarak depolandığı ve ihtiyaç duyulduğunda okunabildiği gelişmiş yapılardır.

Bilgisayarda bilgiler dosyalara kaydedildiği gibi veri tabanlarına da kaydedilir. Bilginin kalıcı olarak saklanabilmesi, istendiğinde çağrılabilmesi, belirli kriterlere göre seçilebilmesi gibi konular oldukça önemlidir. Veri tabanları bilginin saklanması, istendiğinde ve sadece izin verilen yapılarca çağrılabilmesi açısından oldukça gelişmiş yapılardır. Veriler kaydedilirken ve çağrılırken bu işlerin hızlı ve hatasız yapılabilmesi gerektiğinden veri tabanlarında çalışan özel diller geliştirilmiştir. Python ile kullanılacak veri tabanı yapısında sql isimli sorgu dili kullanılmaktadır. Veritabanları yerel bilgisayarlarda tutulabildiği gibi sunucu veya bulut sistemlerde de tutulabilir. Yerel bilgisayar dışında tutulması durumunda bağlantı yapılırken gereken bilgilerin eklenmesi yeterli olacaktır. Bu bölümde yerel veritabanına bağlantı yapacak uygulamalar yapılmıştır.

Veri tabanı ile çalışma mantığı Şekil 14.2'de gösterildiği gibi beş aşamadan oluşmaktadır.

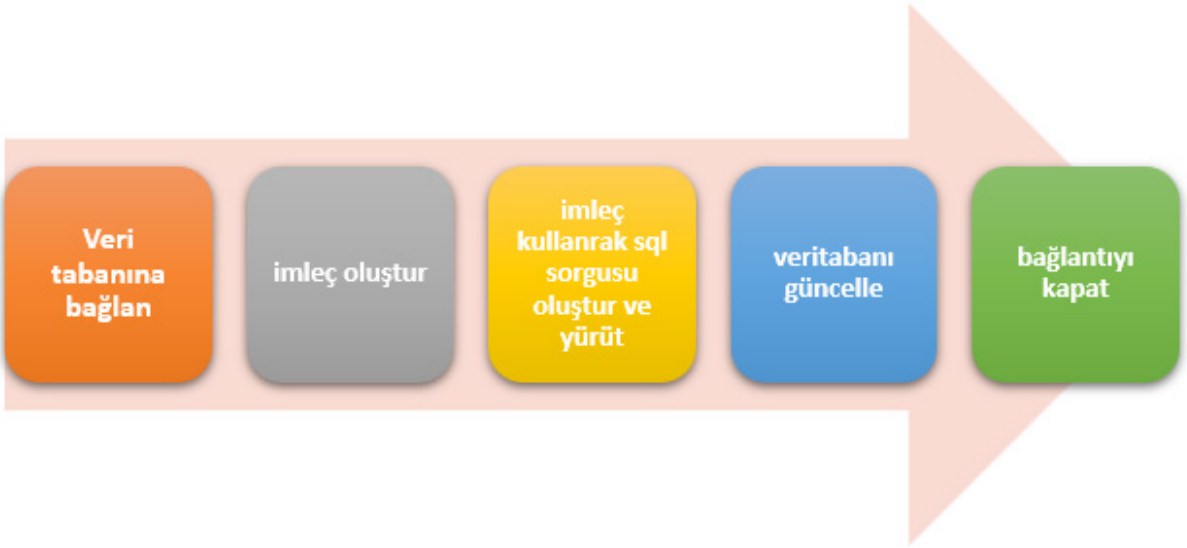
Birinci aşama veritabanı bağlantısının yapılmasıdır.

İkinci aşama sql kodlarının yürütülmesi için imleç nesnesinin oluşturulmasıdır.

Üçüncü aşama imleç nesnesi ile sql kodlarının yazılması ve çalıştırılmasıdır.

Dördüncü aşama veritabanının güncellenmesidir. Bu aşama sadece veri seçme işleminde kullanılmamaktadır.

Beşinci aşama veri tabanı bağlantısının kapatılmasıdır.



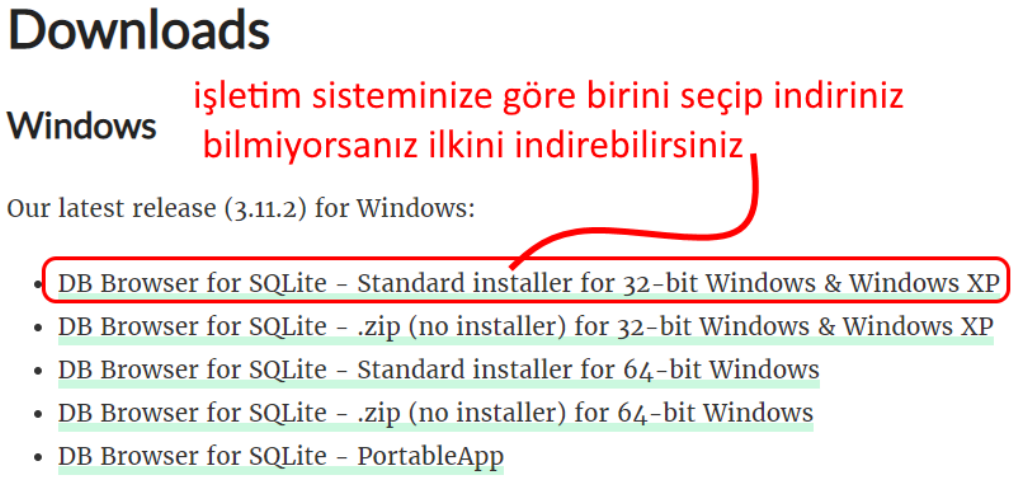
Şekil 14.2: *Veri tabanı ile çalışma süreci*

SQL, veri tabanlarında veri depolamak, işlemek ve almak için standartlaştırılmış bir dildir. Bu dilin öğrenilmesi de ayı bir konu ve uzmanlık alanıdır. Bu nedenle https://www.w3schools.com/sql/ adresinden veya başka Türkçe kaynaklardan sql dilini detaylarıyla öğrenebilirsiniz. Bu bölümde sadece Python ile veri tabanı işlemlerine değinilecektir. Sql içeren kısımlar kısaca anlatılacaktır.

Python ile kullanılabilecek birçok veri tabanı mevcuttur. Bu bölümde yaygın ve kolay olması bakımından sqlite kullanılmıştır. Sqlite'ı Python ile yerel bilgisayarınızda kullanabilmeniz için kurmanız gerekmektedir. Bunun için Şekil 14.3'te gösterildiği gibi https://sqlitebrowser.org/ adresine gidip download düğmesine tıklayınız.

Şekil 14.3: *sqlite web adresi ve indirme bağlantısı*

Ardından işletim sisteminize göre uygun olanı seçip indiriniz. Dilerseniz Şekil-14.4'te gösterildiği gibi ilk seçeneği indirip kurarsanız sorun yaşamadan kullanabilirsiniz.



Şekil 14.4: *sqlite Windows işletim sistemi indirme seçenekleri*

İndirdikten sonra dosyayı çalıştırıp kurulum adımlarını takip ederek uygulamayı bilgisayarınıza yükleyiniz. Artık sqlite ile çalışmaya hazırsınız.

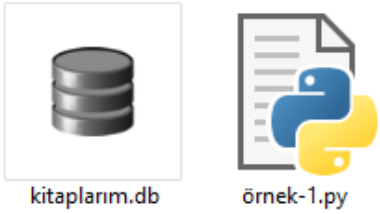
14.1. Sqlite Veritabanı ve Tablo Oluşturma

Python'da Sqlite işlemlerini çalıştırmak için sqlite3 isimli modül kullanılmaktadır. Bu nedenle **import sqlite3** komutunu kullanarak modülü programa dahil ediyoruz. Böylece sql işlemleri için ihtiyaç duyacağımız tüm fonksiyonları kullanabileceğiz. Veri tabanı işlemlerinde öncelikle veri tabanı oluşturulur. Veri tabanı var ise veri tabanına bağlanma işlemine geçilir. **sqlite3.connect(veritabanı adı)** komutu ile veri tabanı yok ise oluşturma ve bağlanma, var ise sadece bağlanma işlemi gerçekleştirilir. Bağlantı işleminden sonra veri tabanı işlemleri için imlec adı verilen bir nesne oluşturulur. İmleç oluşturmak için **sqlite3. cursor()** komutu kullanılır. İlk örnek: kitaplarım isimli veri tabanı uygulamasıdır. Bu uygulamada sadece kitaplarım adında bir veri tabanı oluşturacağız.

Örnek 1

Kitaplık isimli veri tabanı oluşturulması

```
import sqlite3
baglantı=sqlite3.connect("kitaplarım.db")
imlec=baglantı.cursor()
baglantı.close()
```



Şekil 14.5: *Örnek 1 kodları çıktısı*

Kodları kaydettiğimiz klasörde kitaplarım.db dosyasının oluştuğu görülecektir.

14.1.1. Tablo Oluşturma

Veri tabanını oluşturduktan sonra veri tabanına tablo ekleme işlemi yapılacaktır. Sql gibi ilişkisel veri tabanlarında veriler tabloların içinde depolanır. Sql dilinde yeni bir tablo oluşturmak için

CREATE TABLE tablo adı (sütun adı1 veritipi, sütun adı2 veritipi,) yapısı kullanılır. Tabloyu excel tablosu gibi düşünebilirsiniz. Tabloyu oluştururken her bir sütun için isim ve ne tür veri tutulacağını belirtmiş oluyoruz. **CREATE TABLE** dan sonra **IF NOT EXISTS** parametresi de kullanılır. Bu parametre eğer tablo yoksa oluştur anlamına gelir. Eğer bu parametre kullanılmaz ise ve tablo daha önce oluşturulmuş ise **OperationalError: table .... already exists** hatası verir. Bu nedenle **CREATE TABLE IF NOT EXISTS** kullanmak hata ile karşılaşmanın önüne geçer. Bu nedenle tablonun sql yapısı:

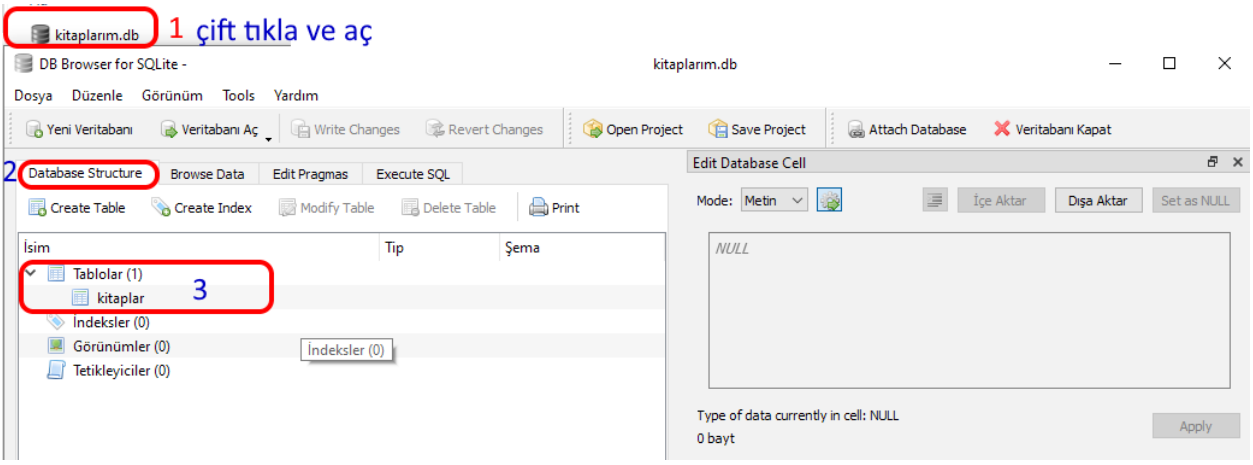
CREATE TABLE IF NOT EXISTS kitaplar (İsim TEXT, Yazar TEXT, Yayınevi TEXT , Sayfa_Sayısı INT) şeklinde olacaktır.

Sql yapısını oluşturduktan sonra bunu imlec nesnesinin execute( ) fonksiyonun içine metinsel ifade olarak ekleyiniz. Son olarak bağlantı nesnesinin commit( ) fonksiyonu ile veritabanı güncellenmektedir. Örnek 2'de kitaplarım veri tabanına tablo ekleme örneği kodları eklenmiştir.

Örnek 2

Veri tabanına tablo ekleme

```
import sqlite3
baglantı=sqlite3.connect("kitaplarım.db")
imlec=baglantı.cursor()
imlec.execute("CREATE TABLE IF NOT EXISTS kitaplar (İsim TEXT, Yazar TEXT, Yayınevi
TEXT, Sayfa_Sayısı INT)")
baglantı.commit()
baglantı.close()
```



Şekil 14.6: Örnek 2 kodları çıktısı

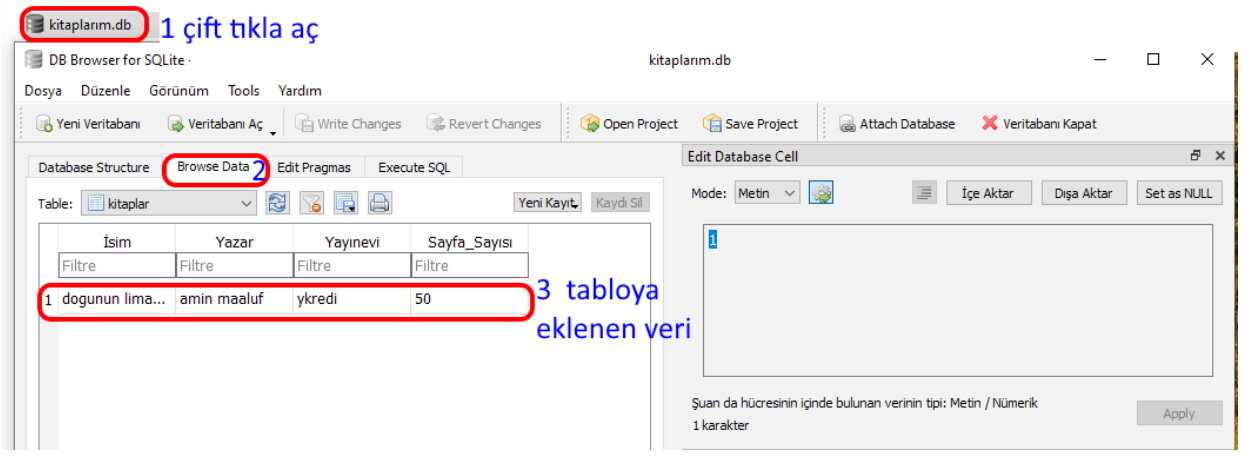
14.2. Veri Ekleme

Sql dilinde tabloya veri eklemek için kullanılan anahtar kelimeler **INSERT INTO tablo_adı (sütun1, sütun 2,) VALUES (değer1, değer2)** veya **insert into tablo_adı (sütun1, sütun 2,) VALUES (değer1, değer2)** şeklinde ifade edilir. Eğer eklenecek verilerin sırası tabloda bulunan sütunlara uyuyor ise **INSERT INTO tablo_adı VALUES (değer1, değer2)** şeklinde de kullanılabilir. Kod yapısı örnek 2- de ki gibidir. Sadece imlec.execute() fonksiyonun içine yazılan sql ifadesi değişmektedir. Ayrıca eklenecek verinin metinsel ifadelerin "tırnak içinde" yazılması gerektiğine dikkat ediniz.

Örnek 3

Tabloya veri ekleme

```
import sqlite3
baglantı=sqlite3.connect("kitaplarım.db")
imlec=baglantı.cursor()
imlec.execute ("INSERT INTO kitaplar VALUES('dogunun limanları','amin maaluf',
'ykredi',50)")
baglantı.commit()
baglantı.close()
```



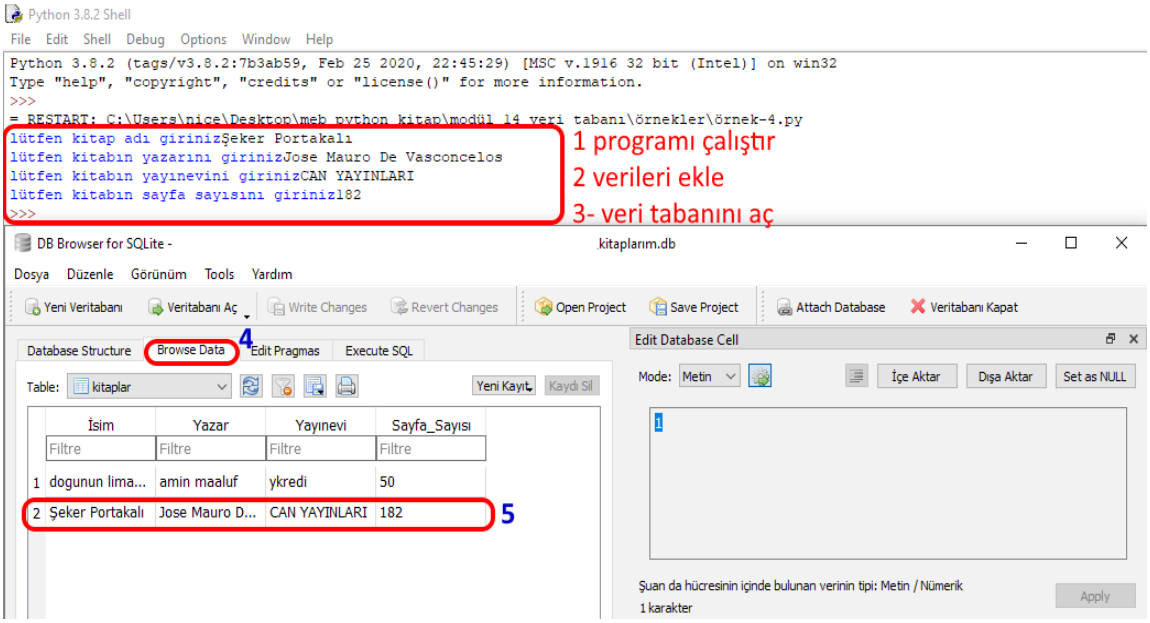
Şekil 14.7: *Örnek 3 kodları çıktısı*

Veri tabanına veri eklendikten sonra kullanıcıdan veri isteyerek tabloya nasıl veri ekleneceği Örnek 4'te gösterilmiştir. Burada dikkat edilecek nokta veriler değişkenlerden alınacağı için direkt olarak values parametresinden sonra kullanılmayacak olmasıdır. "INSERT INTO kitaplar VALUES(?,?,?,?)", (kitap_adı, kitap_yazar, kitap_yayınevi, kitap_sayfa) şeklinde bir yapı kullanılmaktadır. Bu yapı print( ) fonksiyonun kullanımına benzemektedir.

Örnek 4

Kullanıcıdan veri alarak tabloya veri ekleme

```
import sqlite3
def veri_ekle(kitap_adı,kitap_yazar,kitap_yayınevi,kitap_sayfa):
    baglantı=sqlite3.connect("kitaplarım.db")
    imlec=baglantı.cursor()
    imlec.execute ("INSERT INTO kitaplar VALUES(?,?,?,?)",(kitap_adı,kitap_
yazar,kitap_yayınevi,kitap_sayfa))
    baglantı.commit()
    baglantı.close()
kitap_adı=input("lütfen kitap adı giriniz")
kitap_yazar=input("lütfen kitabın yazarını giriniz")
kitap_yayınevi =input("lütfen kitabın yayınevini giriniz")
kitap_sayfa= int(input("lütfen kitabın sayfa sayısını giriniz"))
veri_ekle(kitap_adı,kitap_yazar,kitap_yayınevi,kitap_sayfa)
```



Şekil 14.8: Örnek 4 kodları çıktısı

Sizler programı birkaç defa çalıştırarak farklı kayıtlar oluşturunuz. Böylece sorgu yapacağınız zaman fazla sonuç elde edebilirsiniz.

14.3. Veri Seçme (Çekme)

Sql dilinde veri çekmek (tablodan veri almak) için kullanılan anahtar kelimeler **SELECT sütün1, sütun,2 FROM tablo_adı** şeklindedir. Bu yapıda tabloda belirtilen sütunlar veritabanından çıktı olarak üretilir.

SELECT * FROM tablo_adı Bu yapıda tabloda bulunan tüm sütunlar veritabanından çıktı olarak üretilir.

SELECT * FROM tablo_adı WHERE belirtilen koşullar Bu yapıda tabloda bulunan ve belirtilen koşullara uyan tüm sütunlar veritabanından çıktı olarak üretilir.

SELECT sütun1 FROM tablo_adı WHERE belirtilen koşullar Bu yapıda belirtilen tabloda bulunan, belirtilen koşullara uyan sütun1 verileri veritabanından çıktı olarak üretilir.

Birkaç örnek üzerinden konu pekiştirilecektir. Örnek 5'te kitaplar tablosundaki tüm verileri çekip ekranda yazdıran uygulama bulunmaktadır.

Örnek 5

Tablodan tüm verileri çekmek

```
import sqlite3
baglantı=sqlite3.connect("kitaplarım.db")
imlec=baglantı.cursor()
imlec.execute (" SELECT * FROM  kitaplar")
gelen_veri_listesi=imlec.fetchall()
print(type(gelen_veri_listesi))
print(gelen_veri_listesi)
baglantı.close()
<class 'list'>
[('dogunun limanları', 'amin maaluf', 'ykredi', 50), ('Şeker Portakalı', 'Jose Mauro De Vasconcelos', 'CAN YAYINLARI', 182)]
```

Örnek 5'te görüldüğü gibi veritabanından gelen veriler liste veri tipinde gelmektedir. Gelen veriler üzerinde liste işlemleri yapılabilmektedir.

Bu aşamada belirli sütunları veritabanından çeken uygulama yapılması faydalı olacaktır. Örnek 6'da kitaplar tablosundaki kitap isimlerini veri tabanından çeken uygulama bulunmaktadır:

Örnek 6

Tablodan kitap adlarını çekmek

```
import sqlite3
bağlantı=sqlite3.connect("kitaplarım.db")
imleç=bağlantı.cursor()
imleç.execute (" SELECT İsim FROM  kitaplar")
gelen_veri_listesi=imleç.fetchall()
print(gelen_veri_listesi)
print("kitaplığınızda",len(gelen_veri_listesi),"adet kitap bulunmaktadır.")
bağlantı.close()
[('dogunun limanları',), ('Şeker Portakalı',)]
kitaplığınızda 2 adet kitap bulunmaktadır.
```

Belirli kriterlere uyan verileri tablodan çeken uygulama yapmak konun daha iyi kavranması açısından yararlı olacaktır. Örnek 7'de kitaplar tablosunda bulunan ve yayınevi olarak yapı kredi olan verileri veri tabanından seçen uygulama bulunmaktadır.

Örnek 7

Tablodan belirtilen kritere uygun verileri çekme uygulaması

```
import sqlite3
bağlantı=sqlite3.connect("kitaplarım.db")
imleç=bağlantı.cursor()
imleç.execute (" SELECT * FROM  kitaplar WHERE Yayınevi='ykredi'")
gelen_veri_listesi=imleç.fetchall()
print("sorgunuz ile eşleşen toplam = ",len(gelen_veri_listesi),"kayıt bulunmuştur")
print(gelen_veri_listesi)
bağlantı.close()
sorgunuz ile eşleşen toplam =  1 kayıt bulunmuştur
[('dogunun limanları', 'amin maaluf', 'ykredi', 50)]
```

14.4. Veri Güncelleme

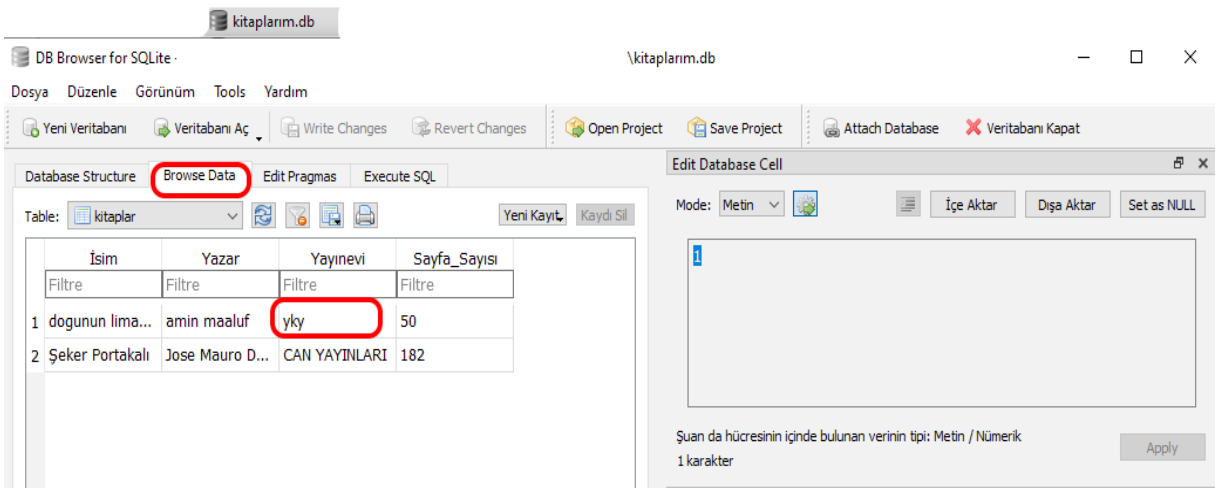
Sql dilinde veri güncelleme işlemi: **"UPDATE tablo_adı SET yeni_değer WHERE değişecek değer"**

Örnek 8'de kitaplar tablosunda yayın evi değerini ykredi'den yky'e değiştiren uygulama bulunmaktadır.

Örnek 8

Tabloda veri güncelleme uygulaması

```
import sqlite3
baglantı=sqlite3.connect("kitaplarım.db")
imlec=baglantı.cursor()
imlec.execute (" UPDATE kitaplar SET Yayınevi= ? WHERE Yayınevi=?",('YKY ', ykredi))
baglantı.commit()
baglantı.close()
```



Şekil 14.9: Örnek 9 kodları çıktısı

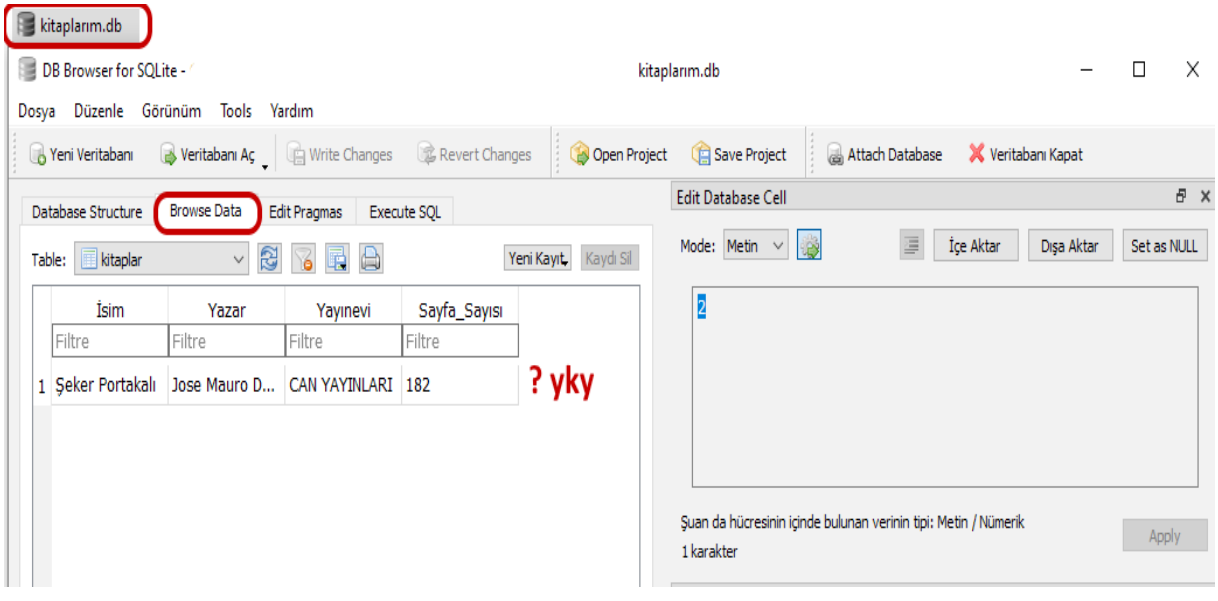
14.5. Veri Silme

Sql dilinde veri silme işlemi: **"DELETE FROM tablo_adı WHERE koşulumuz"** koşuluna uyan kayıtlar veri tabanından silinir. Örnek 9'da yky yayınevine ait kitapları silen uygulama bulunmaktadır.

Örnek 9

Tablodan veri silme uygulaması

```
import sqlite3
baglantı=sqlite3.connect("kitaplarım.db")
imlec=baglantı.cursor()
imlec.execute (" DELETE FROM kitaplar WHERE Yayınevi='yky'")
baglantı.commit()
baglantı.close()
```



Şekil 14.10: Örnek 10 kodları çıktısı

Örnek 10'da kullanıcının gireceği kitap ismine uyan verileri silen uygulama bulunmaktadır.

Örnek 10

Tablodan kullanıcının girdiği kriterlere uyan verileri silme uygulaması

```
import sqlite3
baglantı=sqlite3.connect("kitaplarım.db")
imlec=baglantı.cursor()
kitap_adı=input("silmek istediğiniz kitap adını giriniz")
imlec.execute (" DELETE FROM kitaplar WHERE İsim=? ",(kitap_adı,))
baglantı.commit()
baglantı.close()
```

14.6. Örnek Proje Uygulaması

Veri tabanı işlemleri bölüm bölüm uygulandı. Örnek 11'de proje üzerinde tüm veritabanı işlemlerini içeren uygulama bulunmaktadır. Örnek proje için bir firmanın olduğu varsayılmaktadır. Bu firma için bir veri tabanı oluşturulması ve personel kayıtları için de bir tablo oluşturulması gerekmektedir. Ardından personel tablosuna veri ekleme, seçme, silme ve güncelleme işlemlerinin yapabileceği bir uygulama geliştirilmiştir. İlk aşamada veri tabanı ve tablo oluşturulacaktır. Personel tablosunda kimlik numarası, ad soyad, telefon ve e posta, rolü, çalışılan birim bilgileri kaydedilecektir.

Örnek 11

Firma veri tabanı ve personel tablosu oluşturma

```
import sqlite3
bağlantı=sqlite3.connect("firma.db")
imleç=bağlantı.cursor()
imleç.execute("CREATE  TABLE IF NOT EXISTS personel (kimlik_no INT ,ad_soyad TEXT,
telefon_no TEXT,e_posta TEXT,rolu TEXT,çalışılan_birim TEXT)")
bağlantı.commit()
bağlantı.close()
```

Veri tabanı ve tablo oluşturulduktan sonra veri ekleme, güncelleme, sorgulama ve silme işlemlerinin tek bir program ile yapabileceği Python uygulamasının kodları Örnek 12'de gösterilmiştir.

Örnek 12

Tablo işlemleri uygulaması

```
import sqlite3
baglantı=sqlite3.connect("firma.db")
imlec=baglantı.cursor()
def veri_ekle (veri):
    bilgiler=veri.split(",")
    imlec.execute("INSERT INTO personel
VALUES(?,?,?,?,?,?)",(bilgiler[0],bilgiler[1],bilgiler[2],bilgiler[3],bilgiler[4],
bilgiler[5]))
    baglantı.commit()

def veri_getir (sorgu_metni):
    imlec.execute(sorgu_metni)
    sonuc=imlec.fetchall ()
    print("veri tabanında buluna kayıtlar...")
    for satır in sonuc:
        print(satır)
    print("toplam ",len(sonuc),"adet kayıt listelenmiştir.")

def veri_sil(sorgu_metni):
    imlec.execute(sorgu_metni)
    baglantı.commit()

def veri_güncelle(sorgu_metni):
    imlec.execute(sorgu_metni)
    baglantı.commit()
```

```
while True:
    secim=input('''
|--------------------------------------------|
| yapmak istediğiniz işlemi seçiniz          |
| 1- veri ekleme işlemi                      |
| 2- veri sorgulama işlemi                   |
| 3- veri güncelleme işlemi                  |
| 4- veri silme işlemi                       |
| E- programı sonlandırma işlemi             |
|--------------------------------------------|
''' )
    if secim=="1":
        veri=input('''kimlik numarası, ad soyad, telefon, e posta, rolü,
        çalışılan_birim bilgilerini araya virgül koyarak yazınız''')
        veri_ekle (veri)
    elif secim=="2":
        sorgu=input("lütfen sorgu yapmak için gereken sql metnini giriniz")
        veri_getir (sorgu)
    elif secim=="3":
        sorgu=input("lütfen veri güncelleme yapmak için gereken sql metnini
giriniz")
        veri_güncelle(sorgu)
    elif secim=="4":
        sorgu=input("lütfen sorgu yapmak için gereken sql metnini giriniz")
        veri_sil (sorgu)
    elif secim=="E":
        baglantı.close()
        break
    else:
        print("listede olmayan bir seçim yaptınız")
print("program sonlandırıldı")
```

14.7. Bölüm Sonu Örnekleri

1. Okul adında bir veritabanı oluşturup personel adında tablo oluşturunuz. Örnek tablo alanları ve veri tipleri kimlik_no INT ,ad_soyad TEXT, telefon_no TEXT,e_posta TEXT,rolu TEXT,çalışılan_birim TEXT
2. Okul veri tabanında personel kimlik numarasına göre bilgilerini ekrana getiren programı yazınız.
3. Okul veri tabanında ogrenci adında tablo oluşturunuz.Örnek tablo alanları ve veri tipleri ogrenci_no INT ,ad_soyad TEXT, telefon_no TEXT,e_posta TEXT,sınıfı TEXT,veli_adı TEXT,veli_numarası TEXT
4. Okul veri tabanında öğrenci bilgilerinin tamamını güncelleyebilen programı yazınız.

Cevaplar

1.
```
import sqlite3
baglantı=sqlite3.connect("okul.db")
imlec=baglantı.cursor()
imlec.execute("CREATE  TABLE IF NOT EXISTS personel (kimlik_no INT ,ad_soyad
TEXT, telefon_no TEXT,e_posta TEXT,rolu TEXT,çalışılan_birim TEXT)")
baglantı.commit()
baglantı.close()
```

2.
```
import sqlite3
baglantı=sqlite3.connect("okul.db")
imlec=baglantı.cursor()
kimlik_no=input("bilgilerini görmek istediğiniz personelin kimlik numarasını
giriniz")
sorgu="SELECT * FROM personel WHERE kimlik_no=' "+kimlik_no+" '"
imlec.execute(sorgu)
bilgiler=imlec.fetchall()
print(bilgiler)
baglantı.close()
```

3.
```
import sqlite3
baglantı=sqlite3.connect("okul.db")
imlec=baglantı.cursor()
imlec.execute("CREATE  TABLE IF NOT EXISTS ogrenci  (ogrenci_no INT ,ad_soyad TEXT,  telefon_no  TEXT,e_posta  TEXT,sınıfı  TEXT,veli_adı  TEXT,veli_numarası TEXT)")
baglantı.commit()
baglantı.close()
```

4.
```
import sqlite3
baglantı=sqlite3.connect("okul.db")
imlec=baglantı.cursor()
ogrenci_no=input("bilgilerini görmek istediğiniz öğrencinin numarasını giriniz")
sorgu="SELECT * FROM ogrenci WHERE ogrenci_no=' "+ogrenci_no+" '"
imlec.execute  (sorgu)
bilgiler=imlec.fetchall()
print(bilgiler)
yeni_bilgiler=input('''
lütfen yeni bilgileri araya virgül koyarak giriniz
ogrenci_no, ad soyad, telefon no, e posta, sınıfı,veli adı, veli telefon numarası ''')
liste=yeni_bilgiler.split(",")
sorgu="UPDATE ogrenci SET ogrenci_no =?,ad_soyad=?,telefon_no=?,e_posta=?,sınıfı=?,veli_adı=?,veli_numarası=? WHERE ogrenci_no=? "
imlec.execute(sorgu,(liste[0],liste[1],liste[2],liste[3],liste[4],liste[5],liste[6],ogrenci_no))
baglantı.commit()
baglantı.close()
```